

## CLAIMS:

1. A method for building a target operating system from a source operating system, comprising:

creating a data object for each of multiple components of the source operating system;

generating a dependency model using the data objects;

identifying features in the source operating system to be included in the target operating system;

selecting components necessary to support the identified features by tracing dependencies according to the dependency model; and

linking the selected components to build the target operating system.

2. The method as recited in claim 1, wherein the data object comprises:

an Export field for storing data to identify component outputs;

a Hard References field for storing data to identify component inputs that are necessary for the operation of the target operating system; and

a Soft References field for storing data to identify component inputs that are not necessary for operation of the target operating system.

3. The method as recited in claim 1, wherein the data object further comprises an Independent Links field for storing data to identify Exports, Hard References and Soft References to or from a table entry.

4. The method as recited in claim 1, wherein the data object further comprises a Name and a Type.

5. The method as recited in claim 4, wherein the data object further comprises a Type designated as "Choice," and wherein the Choice data object comprises at least two Independent Links, each Independent Link being an alternative choice of configurations.

6. The method as recited in claim 1, wherein the generating a dependency model further comprises generating a model by connecting each unresolved data object Reference to an Export that resolves the Reference until no more unresolved References can be resolved.

7. The method as recited in claim 6, wherein only Hard References are resolved.

8. The method as recited in claim 1, further comprising:  
creating a data object for each of multiple features of the source operating system;  
and  
including the feature data objects in the dependency model.

9. The method as recited in claim 1, wherein the selecting components further comprises:

selecting a set of one or more components required to support a feature to be included in the target operating system;

identifying a set of data objects that represent the set of components;

for each unresolved Reference in each data object in the set of data objects, tracing the unresolved Reference to a data object Export that resolves the Reference; and

selecting the component that corresponds to the data object that contains the Export.

10. The method as recited in claim 9, wherein the tracing and selecting are only performed for References that are Hard References.

11. One or more computer-readable media comprising computer-executable instructions for performing the method recited in claim 1.

12. A method for generating a dependency model for an operating system, comprising:

creating a component data object for each of multiple components of the operating system, each component data object identifying Exports, Hard References and Soft References for an operating system component associated with the component data object; and

connecting each Reference to an Export that resolves the Reference.

13. The method as recited in claim 12, further comprising creating one or more feature data objects, each feature data object identifying a feature of the operating system and referencing one or more component data objects.

14. The method as recited in claim 12, wherein creating a component data object further comprises associating an object Name and object Type with the component data object.

15. The method as recited in claim 12, wherein the connecting each Reference further comprises connecting each Hard Reference to an Export that resolves the Hard Reference.

16. The method as recited in claim 12, wherein the creating a component data object further comprises identifying one or more Independent Links, each Independent Link identifying one or more Exports from and/or one or more References to a table entry.

17. One or more computer-readable media comprising computer-executable instructions for performing the method as recited in claim 12.

**18.** A method for building a target operating system from a source operating system, comprising:

identifying features in the source operating system to include in the target operating system;

utilizing a dependency model that maps operating system features to operating system components to select components from the source operating system that support the identified features; and

linking the selected components to create the target operating system.

**19.** The method as recited in claim 18, wherein the dependency model includes a data object for each component in the source operating system, each data object potentially having one or more Exports that provide data to at least one other component, and one or more References that reference data from at least one other component.

**20.** The method as recited in claim 19, wherein References further comprise Hard data References that are References that reference data from at least one other component that is critical to the function of the component represented by the data object in which the Hard Reference is included, and Soft References that are References that reference data that is not critical to the function of the component represented by the data object in which the Soft Reference is included.

21. The method as recited in claim 19, wherein each data object can include one or more Independent Links, each Independent Link potentially having one or more Exports that provide data to at least one other component, and one or more other Hard References that Reference a data entry in a data table, the referenced data being critical to the function of the data object in which the Independent Link is included.

22. The method as recited in claim 18, wherein:

the dependency model includes a data object for each component in the source operating system, each data object potentially having one or more Exports that provide data to at least one other component, and one or more Hard References that use data from at least one other component; and

the utilizing the dependency model to select components from the source operating system further comprises:

identifying a selection of an operating system feature;

identifying one or more first operating system components mapped from the selected feature;

for each identified first operating system component, tracing each Hard Reference from the identified component to a component Export from a second operating system component that resolves the Hard Reference and selecting the second operating system component;

if the second operating system component includes one or more Hard References then, for each Hard Reference, tracing the Hard Reference from the second operating system component to a component Export from a third operating

system component that resolves the Hard Reference and selecting the third operating system component; and

repeatedly tracing Hard References and selecting components in a like manner until all Hard References from all components have been resolved and the components that contain the Hard References and the Exports that resolve them have been selected.

**23.** One or more computer-readable media comprising computer-executable instructions for performing the method recited in claim 18.

**24.** A data object stored on one or more computer-readable media, comprising:

an Export field for storing Export information about data provided from an operating system component represented by the data object; and

a Hard Reference field for storing Hard Reference information about data Referenced by the operating system component represented by the data object, the referenced component being critical to the function of the operating system component represented by the data object;

**25.** The data object as recited in claim 24, further comprising a Soft Reference field for storing Soft Reference information about data referenced by the operating system component represented by the data object, the referenced operating system component being non-critical to the function of the component represented by the data object.

26. The data object as recited in claim 24, further comprising an object Name and an object Type.

27. The data object as recited in claim 24, further comprising an Independent Links field for storing Independent Link information about Exports from and References to a table entry data item, the table entry data item being critical to the function of the operating system component represented by the data object.

28. One or more computer-readable media, comprising computer-executable instructions that, when executed on a computer, perform the following:

identifying a selected feature of a source operating system to be included in a target operating system;

identifying one or more first source operating system components that are associated with the selected feature, the one or more first source operating system components being required for the selected feature to function properly;

selecting the one or more first source operating system components;

identifying one or more second source operating system components that are required for the one or more first source operating system components to function properly;

selecting the one or more second source operating system components; and

linking the selected source operating system components to form a target operating system.



**29.** The one or more computer-readable media as recited in claim 28, wherein:  
each source operating system component is represented by a data object;  
a data object further comprises:  
a Name;  
a Type;  
one or more Exports that identify a source operating system component or components, if any, that requires data from the source operating system component represented by the data object;  
one or more Hard References that identify a source operating system component or components, if any, that provides critical data to the source operating system component represented by the data object; and  
the identifying first and second source operating system components further comprises using a dependency model to trace each Hard Reference from a first data object that represents a first source operating system component to an Export from a second data object that represents a second source operating system component to identify the second source operating system component.

**30.** The method as recited in claim 28, further comprising one or more Soft References, each Soft Reference identifying a source operating system component that provides data to the data object, the data provided to the data object being non-critical to the functioning of the source operating system component represented by the data object.

**31.** The method as recited in claim 28, further comprising one or more Independent Links that each identifies a Reference and/or Export that is contained in a data structure that includes more than one Reference and/or Export.

**32.** The method as recited in claim 28, wherein the dependency model includes all the data objects that represent the source operating system components, and wherein each Hard Reference of each data object is logically connected to a data object Export that resolves the Hard Reference.

**33.** A computer-implemented method for building a target operating system from a source operating system, comprising:

creating a data object for each of multiple components of the source operating system;

generating a dependency model using the data objects;

identifying features in the source operating system to be included in the target operating system;

selecting components necessary to support the identified features by tracing dependencies according to the dependency model; and

linking the selected components to build the target operating system.